SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2015

A *Pausable* File System

Dr James Westland Cain
Quantel Limited

Version 1.0

A *Pausable* File System

- Introduction
- Batch vs Stream processing
- Our Environment
- Protocol Support & Documentation
- Live Demo!
- Consequences & Bigger Picture
- Conclusion

Introduction

- Who Am I?
  - Principal Architect – Software, Quantel
    - Designed many systems for TV & Film production
  - Visiting Researcher, Brunel University
    - Researching Software Development Productivity
  - Occasional - SMB2/3 Implementer!
    - Just implemented enough SMB2/3 to get the feature set we require.
      - As anyone in the Plugfest will testify!

Introduction

- Who are Quantel?
- 40 years old technology innovator in TV & Film
- Built bespoke disk systems for nearly 30 years
- Customers include: ESPN, BBC, Fox Sports, DirectTV, BSkyB, BT Sports, FotoKem, Delux, LightIron, Televisa …
- Many Films you might have seen have passed through our kit (including most 3D films)

Batch vs Stream processing

- Often deal with monolithic files
- Files contain indexes as well as pictures and sound
- Indexes require multiple passes to be generated
- Therefore TV and Film processing is often *batch* based:
  - do one thing,
  - wait,
  - do the next thing etc.

Batch vs Stream Processing II

- Batch mode operations are linear
- Batch mode operations are blocking
- Batch mode operations are O(file duration)
- Consequence:
  - Waiting for multiple chained processes gets worse as program duration increases.

Example - Batch vs Stream Processing

- Reality TV Production (with audience voting)
- Live program airs to Cable & Broadcast
- Web version is an edit of the Live version – removing the voting phone numbers etc.
- Web version often only available many hours later
  - Missing the crucial ‘window’ of viewer interest & advertising revenue!

Batch vs Stream Processing IV

- What if we could ‘stream’ media though batch mode tools?
- Delay from start of TV production to start of Web Streaming would be O(process latency)
- The duration of the program would no longer have any bearing on the delay before the Web version could start

Our Environment

- We have built a File Server that offers an SMB2/3 implementation (in user mode on Windows!)
  - We delegate many operations to the underlying OS – so it's a bit of an *Overlay FS*
- All the files are virtual – so the contents of a file doesn't exist before it is read
  - See 'RESTful file-systems' from SDC 2010

Protocol Support & Documentation

- [MS-SMB2].PDF
- <156> Section 3.2.6.1: Windows clients use a default time-out of 60 seconds.
- So for each SMB2_Read request we could just delay the response for up to a minute.
- So this can slow the throughput of the File Server.
- Is there a better way?

Protocol Support & Documentation II

- 3.2.5.1.5 Handling Asynchronous Responses
- If SMB2_FLAGS_ASYNC_COMMAND is set in the Flags field of the SMB2 header of the response and the Status field in the SMB2 header is STATUS_PENDING, the client MUST mark the request in Connection.OutstandingRequests as being handled asynchronously …

Protocol Support & Documentation III

- If SMB2_FLAGS_ASYNC_COMMAND is set in the Flags field of the SMB2 header and Status is not STATUS_PENDING, this is a final response to a request which was processed by the server asynchronously

Protocol Support & Documentation IV

- <144> Section 3.2.5.1.5: Windows clients extend the Request Expiration Timer for requests being processed asynchronously as follows:
  - registry value ExtendedSessTimeout or,
  - the clients extend the expiration time to four times the value of default session timeout.

Protocol Support & Documentation V

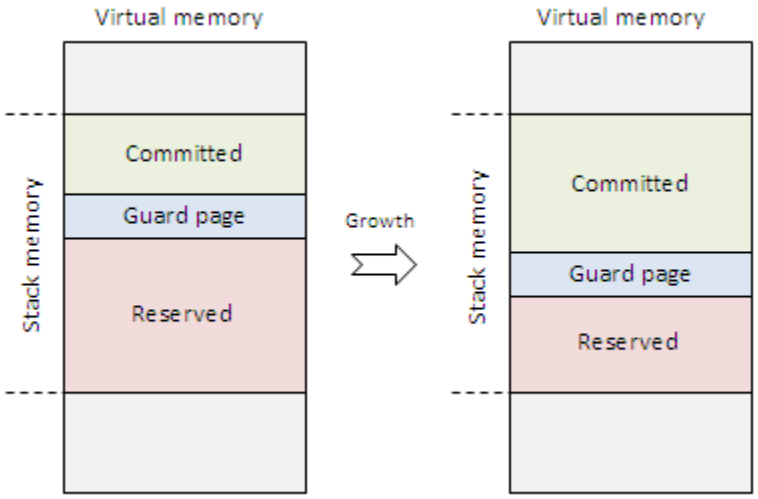
- So by default we can get to 4 minutes
- We can configure the Windows Client to even greater values
- Other OSes will vary
  - That's one of the reasons I'm at the Plugfest!
- The key point is the **Client OS** is doing the work
- The **Client Application** does not know!

Live Demo!

Gulp!

Consequences & Bigger Picture

- Windows call stack growth using guard pages



Consequences & Bigger Picture

- Virtual Files are a bit like Virtual Memory
- We can add ‘guard pages’ to allow for time to create replies when we need the client to not know we haven’t made the file contents yet!
- So the file ‘guard pages’ have the same semantics as Futures in Computer Science.

Consequences & Bigger Picture

- In computer science, **future, promise, and delay** refer to constructs used for synchronization in some concurrent programming languages. They describe an object that acts as a **proxy for a result that is initially unknown**, usually because the computation of its value is yet incomplete.
  - Wikipedia - Futures_and_promises

Consequences & Bigger Picture

- By using this technique with more than one Client Application, we can in effect build Barriers
- In parallel computing, a barrier is a type of synchronization method. A barrier for a group of threads or processes in the source code means any thread/process must stop at this point and cannot proceed until all other threads/processes reach this barrier.

Conclusion

- Techniques to delay read replies allow Application semantics to be varied
  - Without the Application's consent!
- Turning Batch mode operations into Streaming operations allows for massive decreases in wait time for media production
  - Example reduced 3 hours -> 1 minute.

Questions & Comments?

james.cain@quantel.com